

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais
de Eletromiografia de Superfície para
Classificação de Movimentos Utilizando Redes
Neurais Artificiais**

Porto Alegre

2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais de
Eletromiografia de Superfície para Classificação de
Movimentos Utilizando Redes Neurais Artificiais**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Eng. Alexandre Balbinot

Porto Alegre

2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais de
Eletromiografia de Superfície para Classificação de
Movimentos Utilizando Redes Neurais Artificiais**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Trabalho aprovado. Porto Alegre, XX de XXXXXX de 2015:

Prof. Dr. Eng. Alexandre Balbinot
Orientador

Professor
Convidado 1

Professor
Convidado 2

Porto Alegre
2015

*A Gilberto, mi padre, torre de razón y de firme fe;
e a todos aqueles que tomarem interesse neste estudo.*

Agradecimentos

Aos demais colaboradores e pesquisadores do laboratório de Instrumentação Eletro-Eletrônica, em especial Vinicius Cene e Fernanda Trevisol, que desenvolveram a coleta da base de dados utilizada e prestaram auxílio de forma geral.

*Take nothing on its looks;
take everything on evidence.
There's no better rule.*

Charles Dickens, Great Expectations

Resumo

A segmentação de sinais de eletromiografia (EMG) é parte essencial da etapa de pré-processamento em aplicações de reconhecimento de movimentos e controle de próteses. Métodos de segmentação possibilitam a separação automática de trechos de interesse do sinal correspondentes a esforços musculares e o descarte de trechos de sinal com baixa atividade muscular, por exemplo. Neste estudo, quatro métodos para segmentação automática de sinais de EMG de superfície são propostos e implementados em MATLAB. Os métodos foram aplicados nos sinais da base de dados do projeto NinaPro e nos sinais da base de dados adquiridos pelo Laboratório de Instrumentação Eletro-Eletrônica da UFRGS. Redes neurais artificiais (RNAs) foram utilizadas para classificar os sinais segmentados com os quatro métodos entre **XX** movimentos de mão e punho. **TODO: RESULTADOS**.

Palavras-chave: Eletromiografia. Segmentação. Base de dados NinaPro. Redes Neurais Artificiais.

Abstract

TODO: TRADUZIR RESUMO.

Keywords: Eletromiography. Segmentation. MATLAB. NinaPro database. Artificial Neural Networks.

Lista de Figuras

Figura 1 – Soma de potenciais de ação das n fibras de uma unidade motora, formando uma MUAP $h(t)$	15
Figura 2 – MUAPTs de diferentes MUs somam-se para compor o sinal adquirido por um canal de EMG.	16
Figura 3 – Fluxograma representativo do MTD1.	18
Figura 4 – Fluxograma representativo do MTD2.	20
Figura 5 – Fluxograma representativo do MTD3.	24
Figura 6 – Fluxograma representativo do MTD4.	25
Figura 7 – Posicionamento de eletrodos de superfície no braço de um voluntário. .	26
Figura 8 – Trechos do vídeo apresentado aos voluntários na aquisição do IEE. . .	26
Figura 9 – Fluxograma geral para os métodos de segmentação MTD1 - MTD4. . .	27
Figura 10 – Retificação completa de trecho de sinal de eletromiografia.	28
Figura 11 – Normalização de canais de eletromiografia de acordo com seu valor máximo.	28
Figura 12 – Grupamento por DBSCAN dos centros de segmentos obtidos pelos métodos MTD1 e MTD2.	30
Figura 13 – Grupamentos por DBSCAN de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.	31
Figura 14 – <i>Clustering</i> por <i>k-means</i> de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.	32

Lista de Tabelas

Tabela 1	–	Parâmetros utilizados para definir o MTD1.	17
Tabela 2	–	Parâmetros utilizados para definir o MTD2.	19
Tabela 3	–	Parâmetros utilizados para definir o MTD3.	21
Tabela 4	–	Parâmetros utilizados para definir o MTD4.	21
Tabela 5	–	Lista de movimentos de interesse apresentados aos voluntários.	23
Tabela 6	–	Parâmetros ajustáveis para os métodos de segmentação.	29
Tabela 7	–	Combinações de parâmetros utilizados nos métodos de segmentação. . .	33
Tabela 8	–	Número médio de segmentos com diferentes combinações de parâmetros. Base de dados: Ninapro. Método: MTD1.	34

Lista de Abreviaturas e Siglas

BEP	<i>Beginning Extraction Point</i>
DBSCAN	<i>Density-Based Spatial Clustering of Applications with Noise</i>
EEP	<i>Ending Extraction Point</i>
EMG	Eletromiografia
IEE	Laboratório de Instrumentação Eletro-Eletrônica
MU	<i>Motor Unit</i>
MUAP	<i>Motor Unit Action Potencial</i>
MUAPT	<i>Motor Unit Action Potencial Trains</i>
MTD#	<i>Método Número #</i>
NinaPro	<i>Non-Invasive Adaptive Prosthetics project</i>
RNA	Rede Neural Artificial

Sumário

1	INTRODUÇÃO	13
2	REVISÃO BIBLIOGRÁFICA	15
2.1	Sinais de Eletromiografia	15
2.2	Métodos de Segmentação	16
2.2.1	Método 1 - método iterativo utilizando <i>thresholding</i> para detecção de centros de segmentos de comprimento constante (MTD1)	17
2.2.2	Método 2 - método não iterativo utilizando <i>thresholding</i> para detecção de centros de segmentos de comprimento constante (MTD2)	19
2.2.3	Método 3 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando variação total (MTD3)	19
2.2.4	Método 4 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando <i>thresholding</i> (MTD4)	21
2.3	Redes Neurais Artificiais	21
2.4	Bases de Dados Utilizadas	22
2.4.1	Posicionamento de Eletrodos	22
2.4.2	Sistema de Aquisição IEE	22
2.4.3	Movimentos de Interesse	22
3	METODOLOGIA EXPERIMENTAL	27
3.1	Métodos de Segmentação	27
3.1.1	Preprocessamento	27
3.1.1.1	Retificação Completa	27
3.1.1.2	Normalização	27
3.1.2	Parâmetros Ajustáveis	29
3.1.3	Grupamento das Posições de Segmentos de Diferentes Canais Utilizando DBSCAN	30
3.2	Redes Neurais Artificiais	30
3.2.1	Características Utilizadas como Preditores	30
3.2.2	Classes de Movimentos Utilizados como Alvo	31
4	RESULTADOS E DISCUSSÕES	33
4.1	Número de Segmentos Obtidos por Movimento e Combinação de Parâmetros	33
5	CONCLUSÕES	35

6	PROPOSTAS DE FUTUROS TRABALHOS	36
	REFERÊNCIAS	37
	APÊNDICES	39
	APÊNDICE A – FUNÇÃO EM MATLAB PARA MTD1	40
	APÊNDICE B – FUNÇÃO EM MATLAB PARA MTD2	43
	APÊNDICE C – FUNÇÃO EM MATLAB PARA GRUPAMENTO COM DBSCAN	46
	APÊNDICE D – CÓDIGO UTILIZADO PARA DETERMINAÇÃO DE r_{target} DO MTD1	49
	APÊNDICE E – FUNÇÃO EM MATLAB PARA OBTENÇÃO DOS MOVIMENTOS RELACIONADOS AOS SEGMENTOS	50

1 Introdução

Sinais de EMG apresentam crescentes aplicações no controle de próteses mioelétricas. (HARGROVE et al., 2013) mostram o controle de uma prótese de perna de um amputado acima do joelho direito, enquanto (CHU et al., 2007) apresentaram bons resultados de reconhecimento de padrões de EMG para desenvolvimento de uma prótese multifuncional de mão. Em área paralela ao controle de próteses, (PATTICHIS; SCHIZAS; MIDDLETON, 1995) realizam diagnósticos clínicos de desordens neuromusculares com sinais de EMG e redes neurais artificiais.

As principais estratégias para caracterização de sinais de EMG e potenciais de ação das unidades motoras baseiam-se no uso de um método classificador. Métodos de classificação utilizados incluem - entre inúmeros outros - redes neurais artificiais (HUDGINS; PARKER; SCOTT, 1993), classificador Bayesiano (ENGLEHART; HUDGINS, 2003), lógica *fuzzy* (CHAN et al., 2000) e *neuro-fuzzy* (FAVIEIRO; BALBINOT, 2011). Tais sistemas de classificação necessitam, como parte do pré-processamento, segmentar os sinais de EMG adquiridos, para então realizar extração de características dos segmentos como amplitude, número de cruzamentos por zero, coeficientes de autoregressão, transformadas de Fourier e, mais recentemente, transformadas Wavelet (CHU et al., 2007).

Neste trabalho, propõe-se e implementa-se em MATLAB quatro diferentes métodos de segmentação automática para sinais de EMG de superfície. Os primeiros dois métodos (que serão identificados neste estudo pelas abreviações MTD1 e MTD2) tratam da detecção de picos do sinal utilizando *thresholding* e produzem segmentos de comprimento constante centrados nestes picos. O terceiro (MTD3) e quarto (MTD4) métodos utilizam de janela deslizante para identificação de pontos iniciais e finais dos segmentos, produzindo segmentos de comprimento variável.

O primeiro método (MTD1) é baseado no método de segmentação utilizado em (CHAUVET et al., 2001). Trata-se de método iterativo, identificando os picos do sinal a partir de *threshold* de amplitude, segmentando o sinal em janelas de comprimento constante centradas nos picos. O valor de *threshold* para a primeira iteração corresponde ao máximo absoluto do sinal. A cada nova iteração em que não se atinge um número desejado de segmentos, o novo *threshold* é calculado como fração do *threshold* da iteração anterior.

O segundo método (MTD2) é baseado no método de segmentação utilizado em (KATSIKIS et al., 2006). De forma similar ao MTD1, também utiliza *threshold* para detecção de picos do sinal e segmentação com janelas de comprimento constante em torno dos picos. Diferentemente do MTD1, MTD2 não é iterativo, utilizando relação entre valor máximo e comprimento do sinal para cálculo do valor de *threshold*.

O terceiro método (MTD3) é baseado no método de segmentação utilizado em (GUT; MOSCHYTZ, 2000). Uma janela deslizante percorre o sinal e identifica inícios de segmentos quando a declividade média no interior da janela excede determinado valor limite. Os finais dos segmentos são identificados quando a variação total do sinal no interior da janela é inferior a um segundo valor de *threshold*.

O quarto método (MTD4) é baseado no método de segmentação utilizado em (PATTICHIS; SCHIZAS; MIDDLETON, 1995). Os pontos de início do segmento são tais que, em uma janela à esquerda do ponto, o sinal mantém-se abaixo de determinado *threshold*. Os pontos de final de segmento, de forma similar, são tais que, em uma janela à direita do ponto, o sinal mantém-se abaixo do *threshold*.

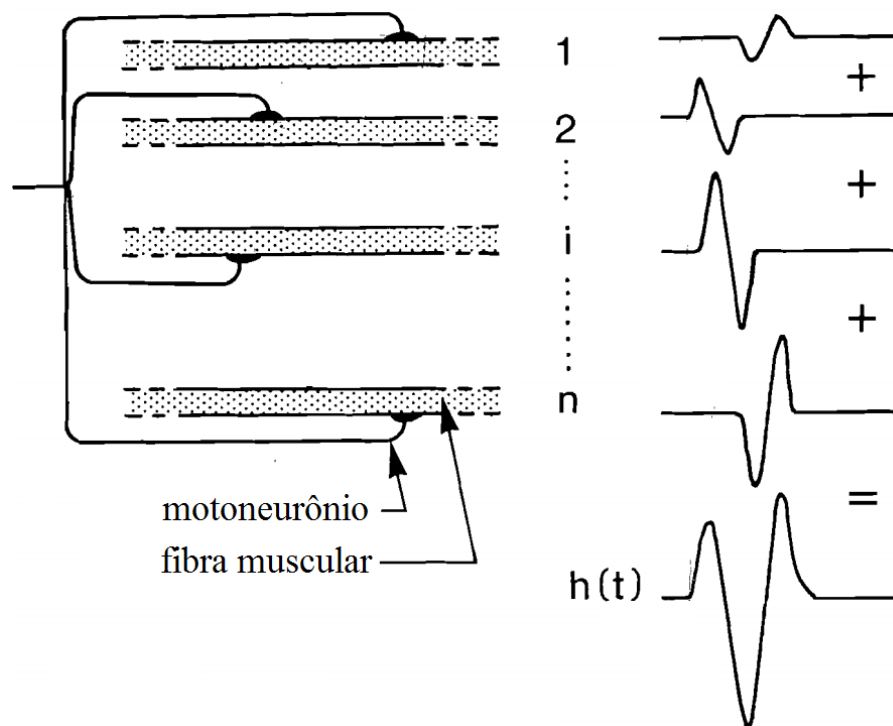
Utilizando valores de RMS, variância e frequência mediana dos segmentos obtidos, redes neurais artificiais (RNAs) serão treinadas para classificar entre movimentos de mão e punho constantes do exercício 1 da base de dados número 2 do projeto NinaPro (GIJSBERTS et al., 2014). Os objetivos finais deste trabalho são a implementação dos métodos de segmentação propostos (para possível uso em trabalhos futuros do laboratório de Instrumentação Eletro-Eletrônica) e fornecer avaliação comparativa entre métodos a respeito da taxa de acerto de classificação obtida com uso de RNA.

2 Revisão Bibliográfica

2.1 Sinais de Eletromiografia

Sinais de EMG podem ser adquiridos por eletrodos posicionados na superfície da pele (eletrodo não invasivo) ou por agulhas introduzidas no tecido muscular (eletrodo invasivo). Os sinais são formados por potenciais de ação de fibras musculares organizadas em unidades funcionais chamadas de “unidades motoras” (MU - *Motor Unit*) (LUCA et al., 2006). Uma unidade motora é composta por um neurônio motor e as fibras musculares que ele inerva, e é a entidade fundamental que controla a ativação de músculos estriados (BUCHTHAL; SCHMALBRUCH, 1980). A soma algébrica dos potenciais de ação de todas as fibras de uma unidade motora é chamada de “potencial de ação da unidade motora”, ou em inglês, MUAP (*Motor Unit Action Potential*) (ALMEIDA; BARRETO, 1997). A Figura 1 apresenta a composição de uma MUAP a partir da soma dos potenciais das fibras de uma unidade motora.

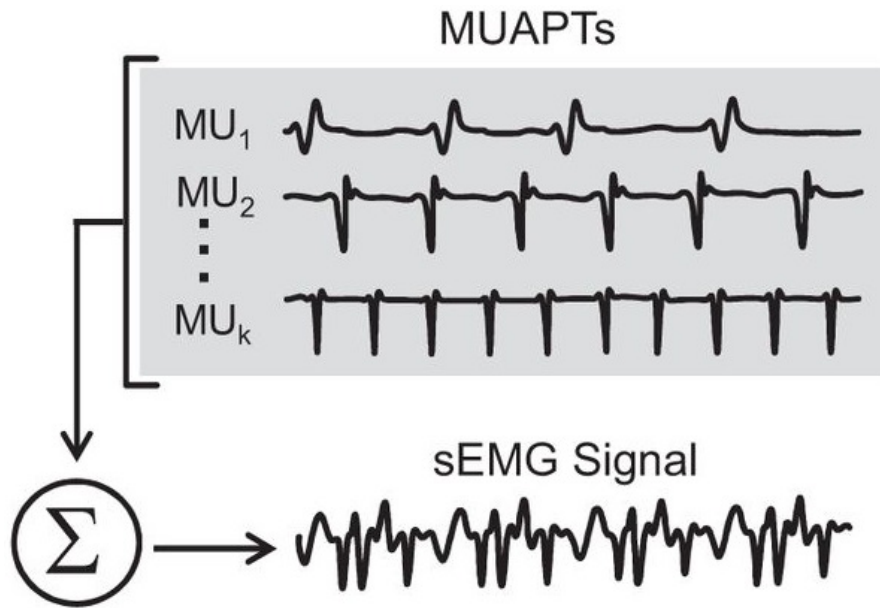
Figura 1: Soma de potenciais de ação das n fibras de uma unidade motora, formando uma MUAP $h(t)$.



Fonte: adaptado de (BASMAJIAN; LUCA, 1985).

Dependendo do método utilizado para aquisição de EMG, é comum a captura da contribuição de mais de uma unidade motora no mesmo canal. A influência de uma unidade motora na amplitude sinal adquirido depende principalmente da distância das fibras musculares ao ponto de aquisição (GERDLE et al., 1999). Sinais de EMG de longa duração são constituídos por sequências temporais de MUAPs, também conhecidas como MUAPTs (*MUAP Trains*). A Figura 2 exemplifica MUAPTs de diferentes MUs que somam-se para formar um sinal de EMG de longa duração.

Figura 2: MUAPTs de diferentes MUs somam-se para compor o sinal adquirido por um canal de EMG.



Fonte: adaptado de (KLINE; LUCA, 2014)

Para maiores detalhes sobre sinais de eletromiografia de superfície, conceitos sobre anatomia e fisiologia sugere-se a consulta de referências clássicas na área e adicionalmente as seguintes referências: (FAVIEIRO, 2009), (LOPES, 2014) e (SCHONS, 2014).

2.2 Métodos de Segmentação

Esta seção descreve os métodos de segmentação desenvolvidos, citando os trabalhos da área que foram utilizados como base teórica para os métodos.

Nota-se que nomes utilizados para variáveis e constantes (por exemplo, sinal a ser segmentado ' x ', *threshold* ' T ', etc.) foram determinados pelo autor deste trabalho, não necessariamente sendo os mesmos utilizados nos trabalhos citados.

Para as definições dos métodos 3 e 4 (MTD3 e MTD4) são utilizados os termos BEP (*beginning extraction point*, ponto inicial de um segmento) e EEP (*ending extraction point*, ponto final de um segmento), também utilizados em (PATTICHIS; SCHIZAS; MIDDLETON, 1995).

2.2.1 Método 1 - método iterativo utilizando *thresholding* para detecção de centros de segmentos de comprimento constante (MTD1)

Este método iterativo é adaptado do método de segmentação utilizado em (CHAUVET et al., 2001). As definições da Tabela 1 serão utilizados para descrever este método.

Tabela 1: Parâmetros utilizados para definir o MTD1.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal a ser segmentado
l	Comprimento desejado para os segmentos
T_k	Valor de <i>threshold</i> para a iteração k
T_{lim}	Valor de limite inferior para o <i>threshold</i>
q	Fração de T_{k-1} para determinação de T_k
N_k	Número total de candidatos para centros de segmentos identificados na iteração k
r_k	Razão entre número de candidatos identificados na iteração k e o comprimento total do sinal
r_{target}	Razão mínima esperada para r_k , utilizada para determinar o final do método

Inicialmente, determina-se o valor de *threshold* T_0 equivalente ao máximo absoluto do sinal a ser segmentado x (equação (2.1)). O valor T_k é atualizado em cada iteração k como sendo uma fração q de T_{k-1} (equação (2.2)). No trabalho de (CHAUVET et al., 2001), este valor q foi empiricamente determinado em 90%.

$$T_0 = \max(x) \quad (2.1)$$

$$T_k = q \times T_{k-1} \quad (2.2)$$

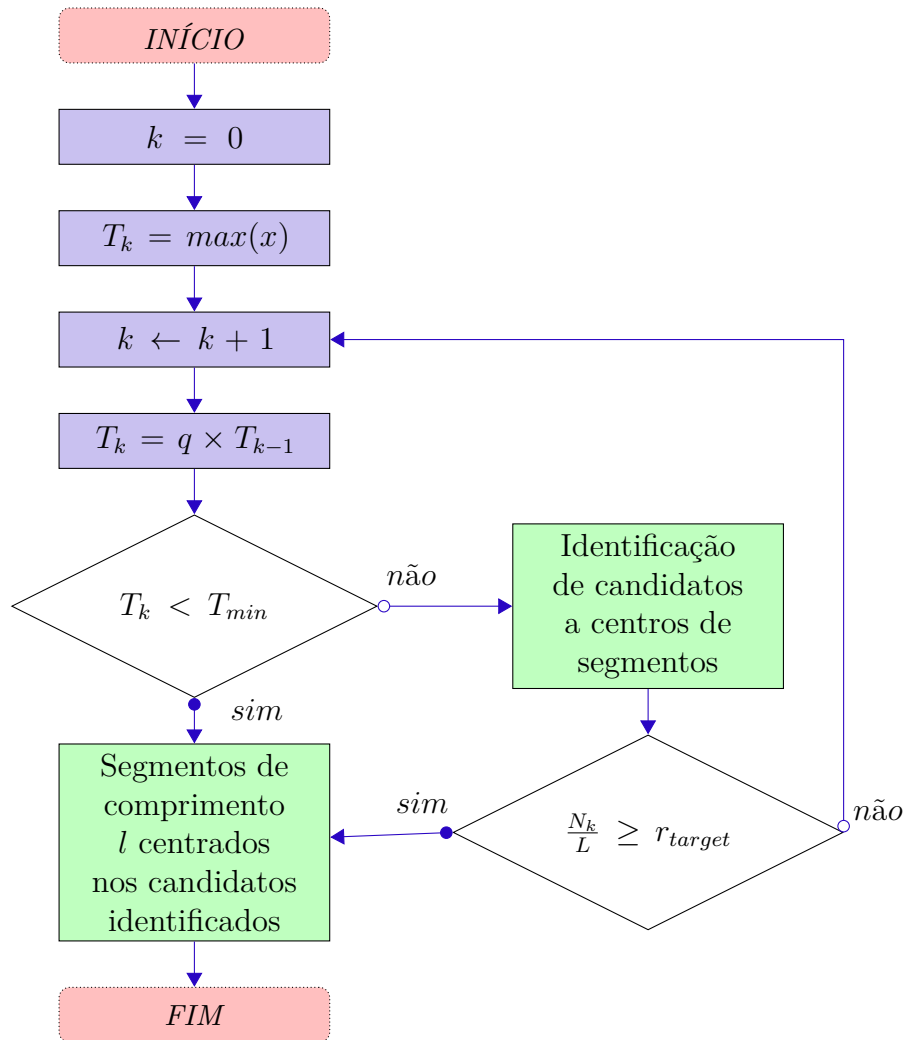
Pontos do sinal acima do valor de T_k são possíveis candidatos para centros de segmentos. Caso exista mais de um possível candidato em uma vizinhança bilateral de l amostras do sinal, apenas o ponto de maior amplitude nesta vizinhança é considerado. Para determinar o final do método, avalia-se a razão r_k entre a quantidade identificada de candidatos N_k e o comprimento total do sinal L (equação 2.3). Caso r_k seja menor que um valor predeterminado r_{target} , calcula-se T_{k+1} para realização da próxima iteração (equação 2.2). Caso r_k seja maior ou igual ao valor predeterminado r_{target} , encerra-se o método e os

segmentos são tomados como janelas de sinal de comprimento l , centradas nos candidatos identificados na última iteração.

$$r_k = \frac{N_k}{L} \quad (2.3)$$

Adicionalmente, o estabelecimento de um valor limite mínimo para *threshold* T_{lim} garante que o método não entre em laço infinito e evita detecção de segmentos em trechos de baixa atividade muscular. Caso o valor de *threshold* T_k para a iteração atual seja inferior a T_{lim} , encerra-se o processo iterativo. O método de segmentação MTD1 é representado pelo fluxograma da Figura 3.

Figura 3: Fluxograma representativo do MTD1.



2.2.2 Método 2 - método não iterativo utilizando *thresholding* para detecção de centros de segmentos de comprimento constante (MTD2)

Este é o método de segmentação utilizado por (KATSIS et al., 2006), que será descrito pelas definições da Tabela 2. Primeiramente, seleciona-se entre dois métodos de cálculo de *threshold* T : ou utiliza-se T como múltiplo da média aritmética do sinal x ; ou T é uma fração do valor máximo do sinal x . No trabalho de (KATSIS et al., 2006) utilizaram a relação do fluxograma da Figura 4 para o cálculo de *threshold* T .

Tabela 2: Parâmetros utilizados para definir o MTD2.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal a ser segmentado
l	Comprimento desejado para os segmentos
T	Valor de <i>threshold</i>
A	Coefficiente utilizado para decisão de método de cálculo de T
B	Múltiplo da média aritmética do sinal x para obtenção de T
C	Fração do valor máximo do sinal x para cálculo de T

De forma similar ao MTD1, os pontos do sinal que tiverem valor acima de T são considerados possíveis candidatos para centros de segmentos. Para os possíveis candidatos que estiverem afastados de uma distância inferior a l , apenas o candidato de maior amplitude é considerado. Em (KATSIS et al., 2006) utilizou-se coeficientes A , B e C respectivamente de 30, 5 e 5, com comprimento l de 121 amostras.

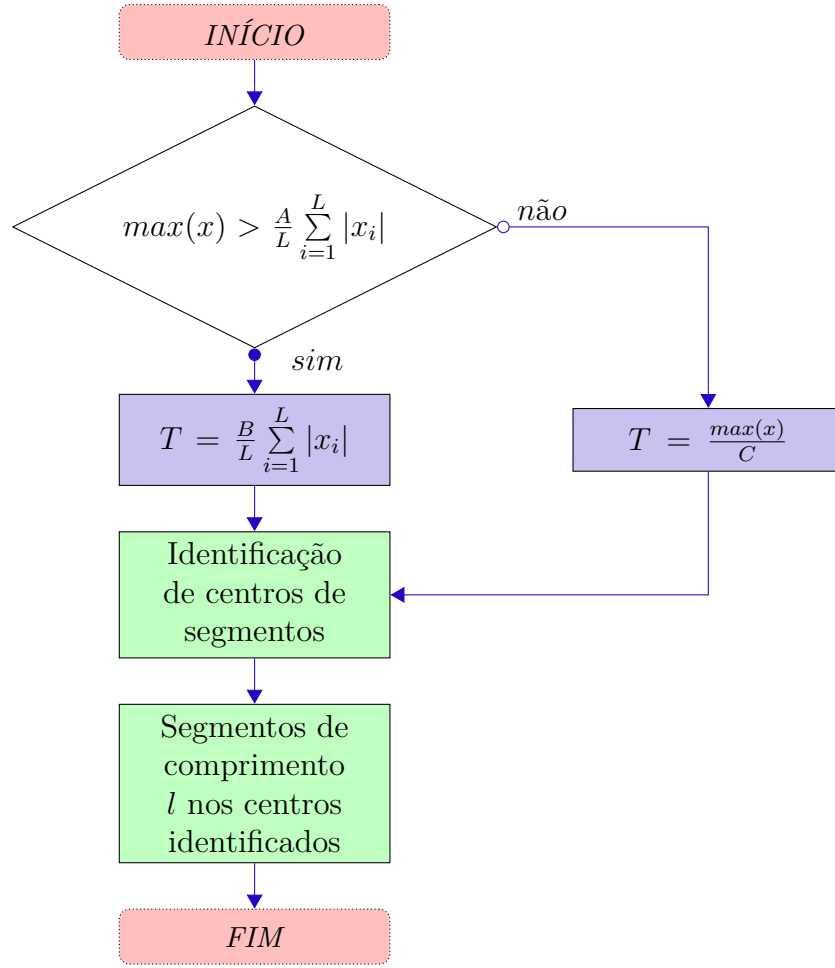
2.2.3 Método 3 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando variação total (MTD3)

Este é o método de segmentação utilizado em (GUT; MOSCHYTZ, 2000). As definições da Tabela 3 serão utilizados para descrever este método.

Uma janela deslizante de comprimento W percorre o sinal da esquerda para a direita. Caso a variação total V (equação (2.4)) do trecho de sinal contido pela janela exceda um limite B (sendo $B > 0$), o ponto mais à esquerda da janela w_0 determina a BEP de um segmento. O EEP do correspondente segmento é então obtido como o ponto mais à direita ($w_0 + W$) da próxima janela na qual a variação total for menor que um limite C (sendo $C < 0$). O MTD3 pode ser representado pelo fluxograma da Figura 5.

$$V = \sum_{i=w_0+1}^{w_0+W} (x_i - x_{i-1}) \quad (2.4)$$

Figura 4: Fluxograma representativo do MTD2.



No método original de (GUT; MOSCHYTZ, 2000), BEPs foram detectadas pelo cálculo da declividade média β no interior da janela (equação (2.5)) nos pontos que excediam um limite B (sendo $B > 0$), e EEPs pelo cálculo do módulo da variação total (equação (2.6)) quando esta era menor que um limite C (sendo $C > 0$).

$$\beta = \frac{1}{W} \sum_{i=w_0+1}^{w_0+W} (x_i - x_{i-1}) \quad (2.5)$$

$$\gamma = \left| \sum_{i=w_0+1}^{w_0+W} (x_i - x_{i-1}) \right| \quad (2.6)$$

Nota-se que $\gamma = |W\beta| = |V|$. O MTD3 proposto, ao utilizar apenas a equação (2.4) para detecção de BEPs e EEPs, explora tal relação para simplificar o método original, com a vantagem adicional de que os limites B e C passam a ser da mesma ordem de grandeza (com o método de (GUT; MOSCHYTZ, 2000), B seria aproximadamente W vezes menor que C).

Tabela 3: Parâmetros utilizados para definir o MTD3.

Nome	Descrição
x	Sinal a ser segmentado
l_{min}	Distância mínima esperada entre BEPs e EEPs de um mesmo segmento
$l_{máx}$	Distância máxima esperada entre BEPs e EEPs de um mesmo segmento
W	Comprimento da janela deslizando utilizada pelo método
w_0	Número da amostra mais a esquerda da janela. Determina a posição instantânea da janela
β	Declividade média do sinal x contido na janela deslizando
B	Valor limite para declividade média que determina um BEP
γ	Variação total do sinal x contido na janela deslizando
C	Valor limite para variação total que determina um EEP

2.2.4 Método 4 - método com janela deslizando para detecção de BEP e EEP de segmentos utilizando *thresholding* (MTD4)

Este é o método de segmentação adaptado de (PATTICHIS; SCHIZAS; MIDDLETON, 1995). As definições da Tabela 4 serão utilizados para descrever este método.

Tabela 4: Parâmetros utilizados para definir o MTD4.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal x
W	Comprimento da janela deslizando utilizada pelo método
T	Valor de <i>threshold</i>

Uma janela deslizando de comprimento W com início em w_0 percorre o sinal da esquerda para a direita. Os BEPs dos segmentos são pontos w_0 tais que o valor máximo do sinal contido pela janela permanece abaixo do valor de *threshold* T . Para sequências de pontos consecutivos que atendam esta especificação, seleciona-se o último ponto (ponto mais à direita). As EEPs são identificadas de forma similar, sendo os primeiros pontos $w_0 + W$ após a identificação de uma BEP nos quais o sinal contido pela janela permanece abaixo do valor de *threshold* T . O fluxograma da Figura 6 representa o MTD4.

2.3 Redes Neurais Artificiais

TODO: explicar RNA e citar exemplos no controle de próteses

2.4 Bases de Dados Utilizadas

Este trabalho faz uso de aquisições para o exercício 1 da base de dados número 2 do projeto NinaPro (GIJSBERTS et al., 2014) e a base de dados em construção realizada pelo Laboratório de Instrumentação Eletro-Eletrônica (IEE). A base de dados do IEE busca replicar os métodos de aquisição utilizados pelo projeto NinaPro, contando com os mesmos movimentos realizados, mesmo posicionamento de eletrodos e mesmo período de amostragem (50 ms).

2.4.1 Posicionamento de Eletrodos

Os sinais de EMG de superfície para ambas as bases são compostos por canais de aquisição de 12 eletrodos posicionados no braço direito de voluntários saudáveis (i.e. não amputados e sem desordens neuromusculares). Os primeiros 8 canais correspondem a eletrodos posicionados de forma a circundar o antebraço e a junção úmero-radial. O eletrodo do canal número 9 é posicionado sobre o músculo flexor superficial dos dedos e o eletrodo do canal 10, em oposição, é posicionado sobre o músculo extensor dos dedos. Os últimos dois eletrodos, 11 e 12, são posicionados sobre o bíceps e o tríceps, respectivamente. A Figura 7 apresenta o posicionamento para os doze eletrodos de superfície e a numeração de seus respectivos canais que compõem os sinais para ambas as bases de dados.

2.4.2 Sistema de Aquisição IEE
















TODO: descrever o *hardware* utilizado na aquisição do IEE.

2.4.3 Movimentos de Interesse

Após o posicionamento de eletrodos, os voluntários sentam-se em frente a um monitor e apoiam o braço direito de forma relaxada sobre uma superfície horizontal próxima à altura do cotovelo. O monitor exibe um vídeo com uma mão direita, digitalmente animada, realizando movimentos de interesse que devem ser mimificados pelos voluntários. A Figura 8 apresenta a mão digitalmente animada realizando alguns dos movimentos de interesse.

Primeiramente, realiza-se um “treinamento” com o voluntário, exibindo o vídeo e apresentando os movimentos que devem ser realizados (sem aquisição de sinal), de modo a reduzir possíveis erros na realização de movimentos para uma segunda exibição, quando os sinais de EMG serão devidamente adquiridos. Cada movimento apresentado em vídeo tem duração de 5 segundos, com um intervalo de 3 segundos de repouso entre movimentos. O voluntário realiza 6 repetições consecutivas para os 17 movimentos descritos na Tabela 5.

Tabela 5: Lista de movimentos de interesse apresentados aos voluntários.

#	Descrição	Imagem demonstrativa	#	Descrição	Imagem demonstrativa	#	Descrição	Imagem demonstrativa
1	Polegar esticado, flexão dos outros dedos.		2	Extensão do indicador e dedo médio, flexão dos outros dedos.		3	Flexão do dedo anelar e mínimo, extensão dos outros dedos.	
4	Polegar para a base do dedo mínimo.		5	Abdução (“afastamento”) de todos os dedos estendidos.		6	Flexão de todos os dedos ao punho.	
7	Extensão do indicador em movimento de “apontar”.		8	Adução (“aproximação”) de todos os dedos estendidos.		9 10	Rotação de punho em torno do dedo médio, dois sentidos.	
11 12	Rotação de punho em torno do dedo mínimo, dois sentidos.		13	Flexão de punho.		14	Extensão de punho.	
15	Desvio radial do punho.		16	Desvio ulnar do punho.		17	Extensão de punho com mão cerrada.	

Fonte: adaptado de (ATZORI et al., 2014).

Figura 5: Fluxograma representativo do MTD3.

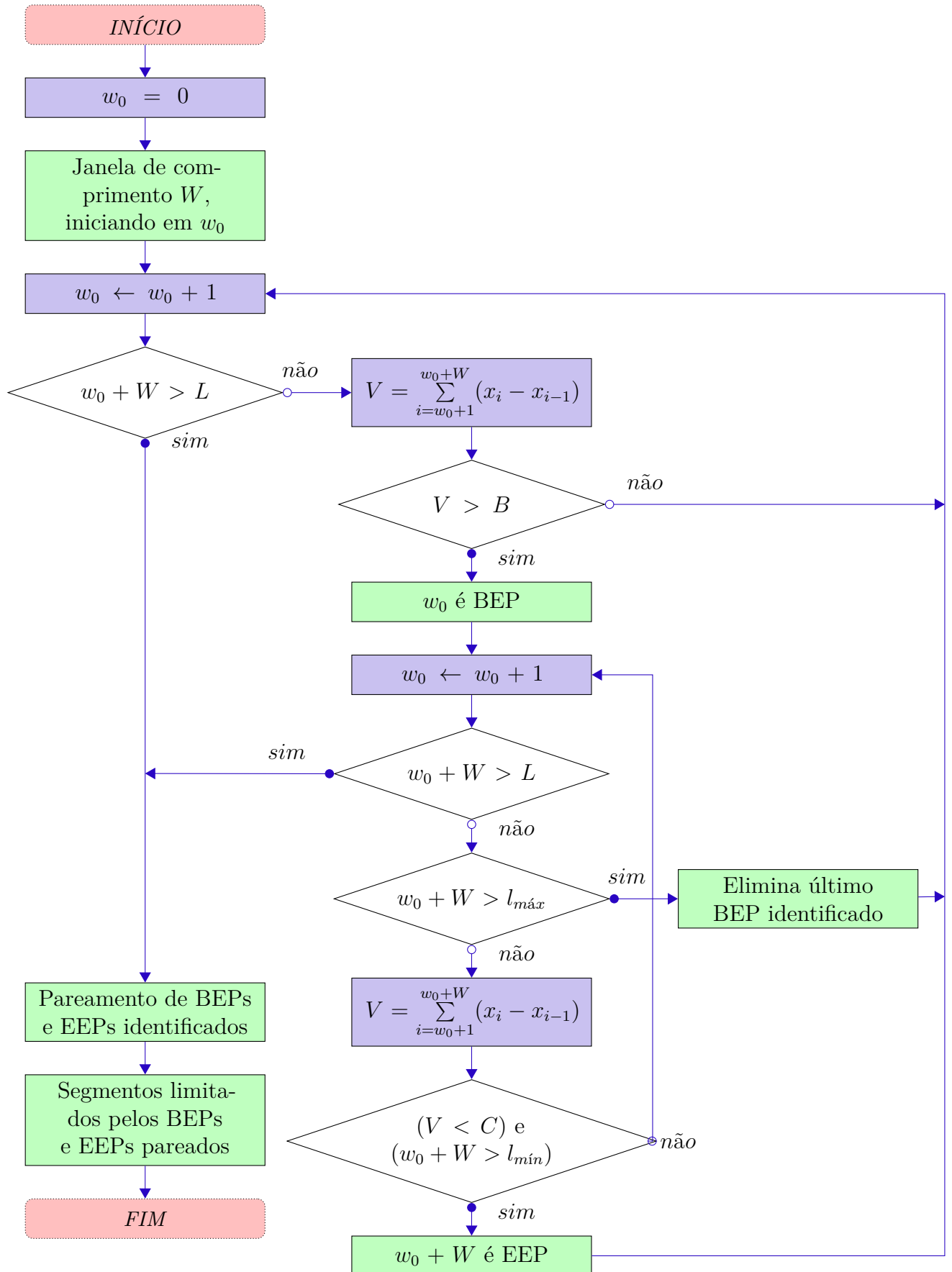


Figura 6: Fluxograma representativo do MTD4.

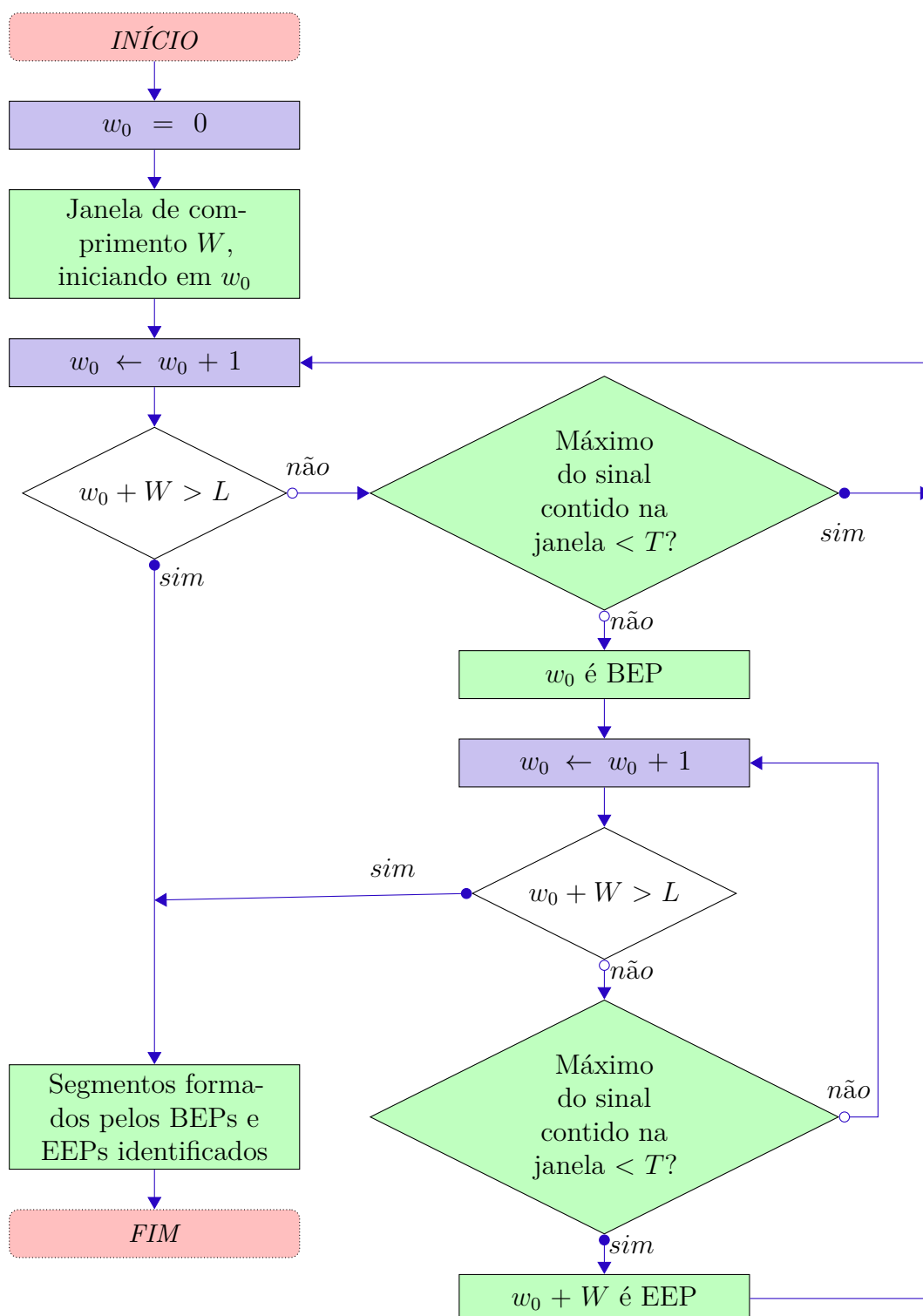
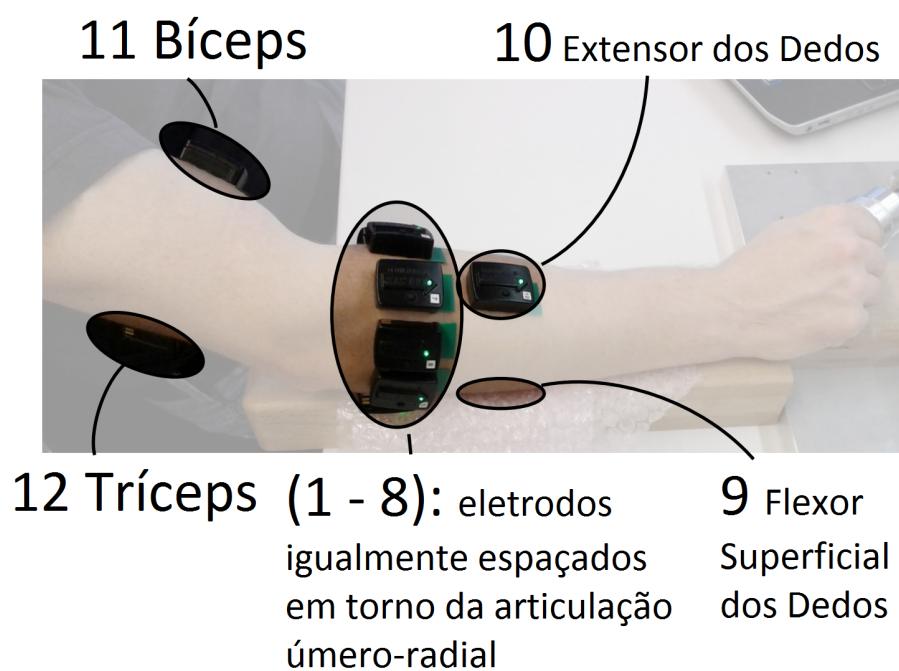
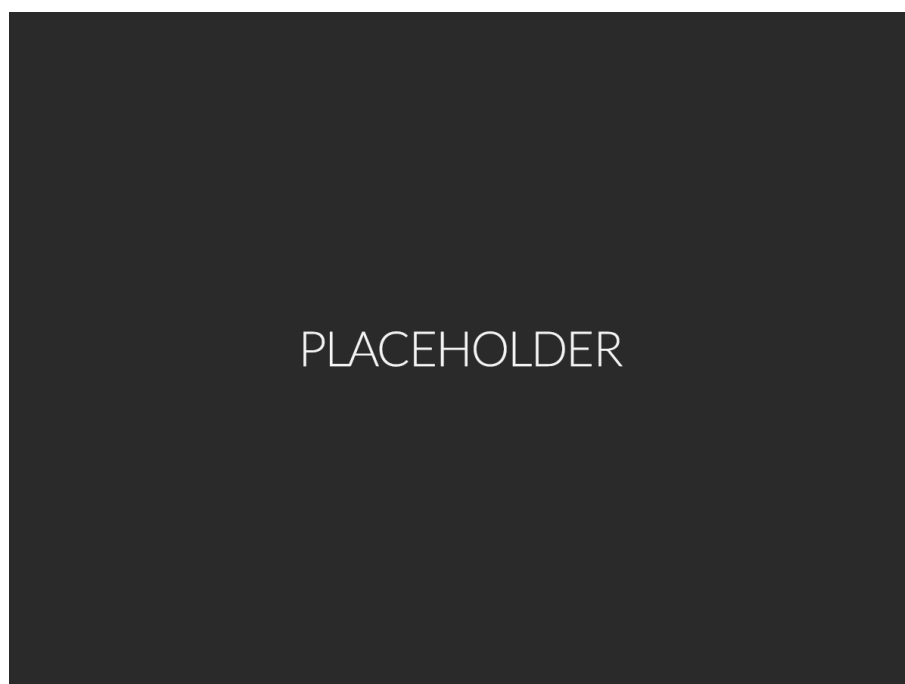


Figura 7: Posicionamento de eletrodos de superfície no braço de um voluntário.



Fonte: adaptado de (GIJSBERTS et al., 2014)

Figura 8: Trechos do vídeo apresentado aos voluntários na aquisição do IEE.

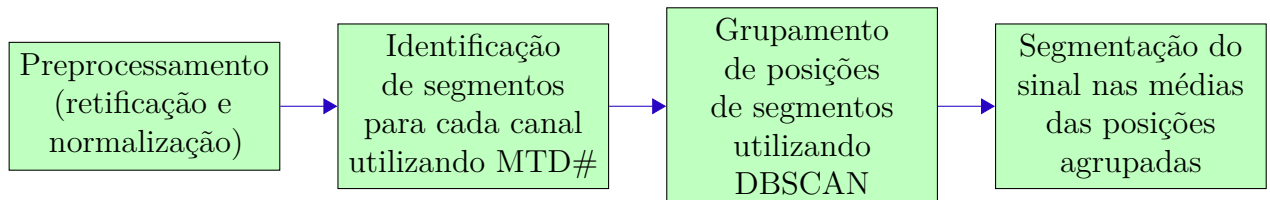


3 Metodologia Experimental

3.1 Métodos de Segmentação

Esta seção descreve a implementação dos métodos de segmentação desenvolvidos. O fluxograma da Figura 9 apresenta de forma resumida os passos comuns aos quatro métodos, que serão explanados nas subseções seguintes. Os códigos em MATLAB criados para os quatro métodos são apresentados nos Apêndices A - D.

Figura 9: Fluxograma geral para os métodos de segmentação MTD1 - MTD4.



3.1.1 Preprocessamento

3.1.1.1 Retificação Completa

Os sinais de eletromiografia para ambas as bases de dados (NinaPro e IEE) são armazenados mantendo sua polaridade original (i.e. amostras do sinal podem assumir valores positivos e negativos). Primeiramente, realiza-se a retificação completa dos sinais tomando o módulo dos valores amostrados (em MATLAB, função *abs()*). A retificação completa do sinal mantém sua energia e é fundamental para a implementação dos métodos de segmentação aqui desenvolvidos. A Figura 10 exemplifica o resultado esperado para a retificação completa de um trecho de sinal de eletromiografia.

3.1.1.2 Normalização

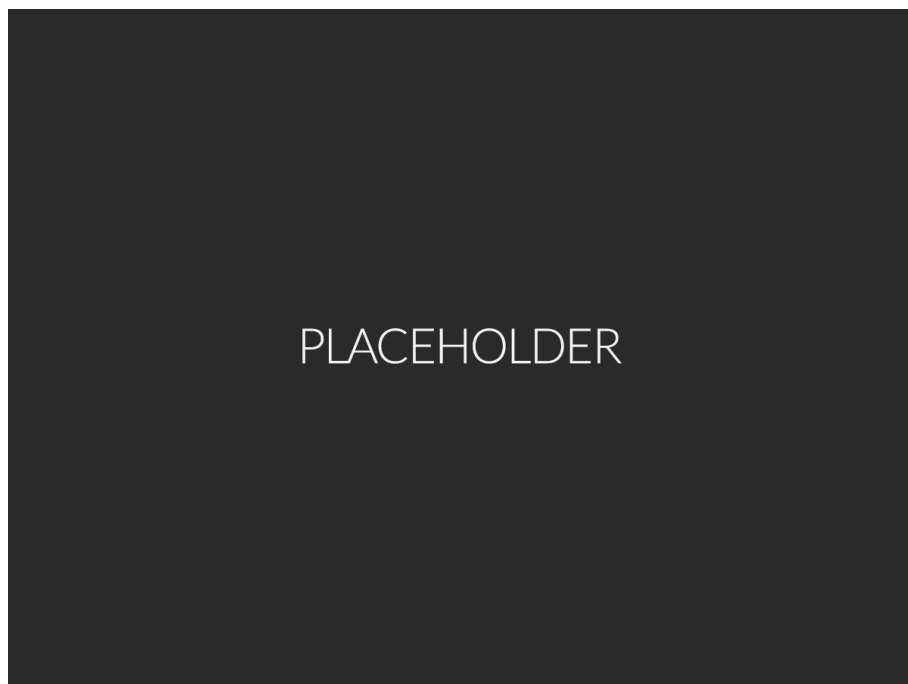
Os sinais para cada canal de aquisição são normalizados de acordo com seu valor máximo, de modo que seu novo valor máximo seja unitário, a partir da equação 3.1, onde x é o sinal original para um canal e x_{norm} é sua versão normalizada. A normalização de canais faz com que os parâmetros utilizados pelos métodos de segmentação sejam relativos ao valor máximo do sinal, possibilitando a implementação para diferentes voluntários. A Figura 11 exemplifica a normalização para três canais de um trecho de sinal já retificado.

Figura 10: Retificação completa de trecho de sinal de eletromiografia.



$$x_{norm} = \frac{x}{\max(x)} \quad (3.1)$$

Figura 11: Normalização de canais de eletromiografia de acordo com seu valor máximo.



3.1.2 Parâmetros Ajustáveis

Cada método de segmentação MTD1 - MTD4 apresenta um conjunto de parâmetros ajustáveis. Tais parâmetros foram descritos anteriormente na seção 2.2. Após investigações iniciais das segmentações obtidas com diferentes valores de parâmetros, fixou-se alguns destes parâmetros e realizou-se uma lista de valores a serem explorados na aplicação de cada método, a fim de obter os parâmetros mais adequados para cada uma das bases de dados, assim como avaliar quais classes de movimentos poderiam ser devidamente segmentados com cada método. A Tabela 6 apresenta os parâmetros ajustáveis de cada método e sua respectiva lista de valores explorados.

Tabela 6: Parâmetros ajustáveis para os métodos de segmentação.

Método	Parâmetros	Valores utilizados	Número total de combinações
MTD1	l	10×10^3	16
	r_{target}	$5, 6 \times 10^{-5}$	
	q	[0.8 0.85 0.9 0.95]	
	T_{lim}	[0.05 0.1 0.15 0.2]	
MTD2	l	10×10^3	27
	A	[20 30 40]	
	B	[2 5 8]	
	C	[2 5 8]	
MTD3	W	TODO	TODO
	B	TODO	
	C	TODO	
MTD4	W	TODO	TODO
	T	TODO	

Para cada movimento de interesse, o voluntário realiza 6 repetições; portanto, espera-se que os métodos de segmentação produzam 6 segmentos por classe de movimento. Para testar as diferentes combinações de parâmetros, utilizou-se os códigos nos apêndices E a H.

Para os métodos que utilizam parâmetro de comprimento de segmento l (i.e. os métodos que produzem segmentos de comprimento de janela constante, MTD1 e MTD2) utilizou-se valor de l como 10×10^3 , que corresponde a 5 cinco segundos de aquisição para ambas as bases de dados (período de amostragem para ambas é de $500\mu s$), sendo a mesma duração dos segmentos de vídeo mimcados pelos voluntários. O valor de MTD1 para r_{target} foi obtido utilizando o código do Apêndice D como valor mínimo da razão entre número de segmentos e comprimento de sinal para a base de dados NinaPro.

3.1.3 Grupamento das Posições de Segmentos de Diferentes Canais Utilizando DBSCAN

Os sinais para ambas as bases de dados são compostos por 12 canais de aquisição. Os métodos de segmentação são implementados individualmente aos doze canais. Para os métodos MTD1 e MTD2, as posições centrais dos segmentos obtidas em cada canal são armazenadas, enquanto que para os métodos MTD3 e MT4 armazena-se as posições de BEPs e EEPs.

TODO: explicar DBSCAN

Figura 12: Grupamento por DBSCAN dos centros de segmentos obtidos pelos métodos MTD1 e MTD2.



3.2 Redes Neurais Artificiais

Esta seção descreve o uso de RNAs para classificação dos segmentos obtidos de acordo com movimentos de interesse. O processo de classificação é representado pelo fluxograma da Figura 14, que será explanado nas subseções seguintes.

3.2.1 Características Utilizadas como Preditores

Os preditores (“entradas”) das RNAs são o valor RMS, a variância e a frequência mediana do espectro de potência dos segmentos de sinal obtidos pelos métodos MTD1 - MTD4. Tais características foram selecionados de acordo com outros trabalhos já realizados

Figura 13: Grupamentos por DBSCAN de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.



no laboratório de Instrumentação Eletro-Eletrônica que obtiveram bons resultado para uso com RNA, como (FAVIEIRO, 2009), (SCHONS, 2014) e (CENE, 2015).

O valor RMS ou *root mean square* de um sinal discreto x de comprimento L (em MATLAB, função $rms()$) é dada pela equação 3.2 e a variância deste sinal (em MATLAB, função $var()$) é dada pela equação 3.3, onde \bar{x} é o valor médio do sinal. A frequência mediana é tal que a soma total da densidade de potência para frequências abaixo da frequência mediana é igual à soma total da densidade de potência para frequências acima da frequência mediana, sendo estimada pela função $medfreq()$ em MATLAB.

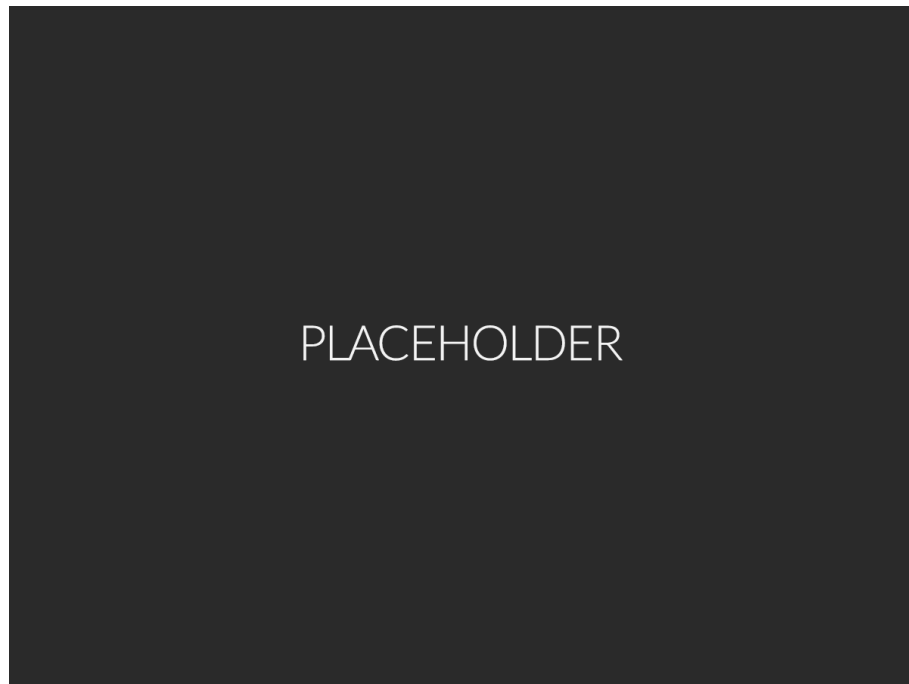
$$rms(x) = \sqrt{\frac{1}{L} \sum_{i=1}^L x_i^2} \quad (3.2)$$

$$var(x) = \frac{1}{L-1} \sum_{i=1}^L (x_i - \bar{x})^2 \quad (3.3)$$

3.2.2 Classes de Movimentos Utilizados como Alvo

Obtém-se a identificação de movimentos associados a cada segmento obtido a partir de sua posição temporal no sinal original, já que durante a aquisição de dados os movimentos são realizados na mesma sequência para todos os voluntários. A base de dados NinaPro apresenta um vetor *stimulus* que indica os instantes em que foram apresentados cada movimento a ser replicado pelo voluntário. A partir deste vetor e das posições dos

Figura 14: *Clustering* por *k-means* de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.



segmentos dentro do sinal original, utiliza-se o código apresentado no Apêndice E para criar a variável que servirá como resposta esperada no treinamento da RNA.

4 Resultados e Discussões

4.1 Número de Segmentos Obtidos por Movimento e Combinação de Parâmetros

Para as diferentes combinações de parâmetros listadas na Tabela 7, as Tabelas ?? a ??, apresentam o número médio de segmentos obtidos para todos os voluntários de cada base de dados utilizada. Como os voluntários realizam 6 repetições de cada movimento, os melhores parâmetros são aqueles que melhor conseguirem identificar este número de segmentos por classe de movimento.

Tabela 7: Combinações de parâmetros utilizados nos métodos de segmentação.

Método#	1		2			3		4
Parâmetro	q	T_{lim}	A	B	C	B	C	T
Índice da Combinação	1	0.80	0.05					
	2	0.85	0.05					
	3	0.90	0.05					
	4	0.95	0.05					
	5	0.80	0.10					
	6	0.85	0.10					
	7	0.90	0.10					
	8	0.95	0.10					
	9	0.80	0.15					
	10	0.85	0.15					
	11	0.90	0.15					
	12	0.95	0.15					
	13	0.80	0.20					
	14	0.85	0.20					
	15	0.90	0.20					
	16	0.95	0.20					
	17							
	18							
	19							
	20							
	21							
	22							
	23							
	24							
	25							
	26							
	27							

TODO: analisar resultados, escolher melhores combinações de movimentos e parâmetros

Tabela 8: Número médio de segmentos com diferentes combinações de parâmetros. Base de dados: Ninapro. Método: MTD1.

Movimento	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Média
1	7,575	7,100	6,950	6,475	6,400	6,500	6,950	6,525	6,900	7,150	8,075	7,000	6,975	6,400	6,625	7,125	7,525	6,9559
2	7,600	7,025	6,925	6,475	6,400	6,525	6,950	6,525	6,900	7,125	8,100	7,025	7,000	6,375	6,550	7,125	7,450	6,9456
3	7,675	7,050	6,925	6,475	6,400	6,500	6,950	6,525	6,925	7,125	8,050	7,000	7,025	6,400	6,575	7,125	7,425	6,9500
4	7,550	7,025	6,900	6,475	6,350	6,475	6,950	6,525	6,900	7,100	8,075	7,025	7,000	6,400	6,525	7,125	7,450	6,9324
5	6,200	6,575	6,550	6,300	6,150	6,300	6,025	6,225	6,850	6,800	7,700	7,025	6,775	6,275	6,375	7,000	7,050	6,5985
6	6,400	6,600	6,525	6,350	6,225	6,275	6,225	6,250	6,875	6,800	7,725	7,025	6,825	6,325	6,400	7,050	7,000	6,6397
7	6,175	6,575	6,500	6,275	6,125	6,275	5,900	6,200	6,825	6,800	7,700	7,025	6,775	6,275	6,350	6,975	6,950	6,5706
8	6,225	6,600	6,500	6,325	6,125	6,275	6,150	6,250	6,875	6,800	7,725	7,025	6,775	6,325	6,400	7,050	6,975	6,6118
9	4,225	5,600	6,150	5,775	5,725	6,000	4,275	5,725	6,350	6,275	7,075	6,425	6,575	6,175	5,675	6,425	6,400	5,9324
10	4,250	5,575	6,150	5,800	5,750	5,975	4,250	5,700	6,350	6,275	7,050	6,425	6,600	6,175	5,650	6,425	6,425	5,9309
11	4,750	5,950	6,225	6,000	5,850	6,100	4,675	5,925	6,500	6,475	7,250	6,675	6,750	6,200	5,925	6,650	6,600	6,1471
12	4,550	5,750	6,150	5,900	5,775	6,050	4,475	5,850	6,425	6,400	7,150	6,500	6,675	6,175	5,825	6,625	6,600	6,0515
13	2,950	5,000	5,650	5,200	5,250	5,750	3,150	4,900	5,600	5,625	6,325	5,700	6,150	5,900	5,200	6,075	6,025	5,3206
14	2,450	4,550	5,300	4,875	4,925	5,625	2,375	4,475	5,400	5,100	5,750	5,200	5,950	5,875	4,725	5,875	5,775	4,9544
15	3,100	5,025	5,650	5,275	5,250	5,775	3,225	4,975	5,725	5,750	6,350	5,800	6,300	5,925	5,250	6,050	6,075	5,3824
16	3,125	5,075	5,725	5,275	5,250	5,800	3,325	5,050	5,775	5,775	6,375	5,800	6,275	5,950	5,275	6,075	6,075	5,4118

Combinação de Parâmetros (Tabela 7)

5 Conclusões

6 Propostas de Futuros Trabalhos

Referências

- ALMEIDA, M.; BARRETO, J. *Filtragem digital de sinais biomédicos*. Universidade Federal de Santa Catarina, Centro Tecnológico., 1997. Disponível em: <<https://books.google.com.br/books?id=L5iqkgEACAAJ>>. Citado na página 15.
- ATZORI, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, Macmillan Publishers Limited SN -, v. 1, p. 140053 EP -, Dec 2014. Data Descriptor. Disponível em: <<http://dx.doi.org/10.1038/sdata.2014.53>>. Citado na página 23.
- BASMAJIAN, J.; LUCA, C. D. *Muscles Alive*. 5. ed. Baltimore: Williams and Wilkins, 1985. Citado na página 15.
- BUCHTHAL, F.; SCHMALBRUCH, H. Motor unit of mammalian muscle. *Physiological Reviews*, American Physiological Society, v. 60, n. 1, p. 90–142, 1980. Citado na página 15.
- CENE, V. H. Upper-limb movement classification based on semg signal validation with continuous channel selection. In: LA-CCI. *Annals of 2nd Latin-American Congress on Computational Intelligence*. Curitiba, PR, 2015. ISBN: 978-85-69972-00-6. Citado na página 31.
- CHAN, F. H. et al. Fuzzy EMG classification for prosthesis control. *IEEE Trans Rehabil Eng*, v. 8, n. 3, p. 305–311, Sep 2000. Citado na página 13.
- CHAUVET, E. et al. A method of emg decomposition based on fuzzy logic. In: *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*. [S.l.: s.n.], 2001. v. 2, p. 1948–1950 vol.2. ISSN 1094-687X. Citado 2 vezes nas páginas 13 e 17.
- CHU, J.-U. et al. A supervised feature-projection-based real-time emg pattern recognition for multifunction myoelectric hand control. *Mechatronics, IEEE/ASME Transactions on*, v. 12, n. 3, p. 282–290, June 2007. ISSN 1083-4435. Citado na página 13.
- ENGLEHART, K.; HUDGINS, B. A robust, real-time control scheme for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, v. 50, n. 7, p. 848–854, July 2003. ISSN 0018-9294. Citado na página 13.
- FAVIEIRO, G.; BALBINOT, A. Adaptive neuro-fuzzy logic analysis based on myoelectric signals for multifunction prosthesis control. In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. [S.l.: s.n.], 2011. p. 7888–7891. ISSN 1557-170X. Citado na página 13.
- FAVIEIRO, G. W. *Controle de uma Prótese Experimental do Segmento Mão-Braço por Sinais Mioelétricos e Redes Neurais Artificiais*. 2009. Trabalho de Diplomação, Curso de Engenharia de Computação, UFRGS. Citado 2 vezes nas páginas 16 e 31.
- GERDLE, B. et al. Acquisition, processing and analysis of the surface electromyogram. In: WINDHORST, U.; JOHANSSON, H. (Ed.). *Modern Techniques in Neuroscience Research*.

- Springer Berlin Heidelberg, 1999. p. 705–755. ISBN 978-3-642-63643-1. Disponível em: <http://dx.doi.org/10.1007/978-3-642-58552-4_26>. Citado na página 16.
- GIJSBERTS, A. et al. Measuring movement classification performance with the movement error rate. *IEEE Transactions on neural systems and rehabilitation engineering*, 2014. Citado 3 vezes nas páginas 14, 22 e 26.
- GUT, R.; MOSCHYTZ, G. S. High-precision emg signal decomposition using communication techniques. *Signal Processing, IEEE Transactions on*, v. 48, n. 9, p. 2487–2494, Sep 2000. ISSN 1053-587X. Citado 3 vezes nas páginas 14, 19 e 20.
- HARGROVE, L. J. et al. Robotic leg control with emg decoding in an amputee with nerve transfers. *New England Journal of Medicine*, v. 369, n. 13, p. 1237–1242, 2013. PMID: 24066744. Disponível em: <<http://dx.doi.org/10.1056/NEJMoa1300126>>. Citado na página 13.
- HUDGINS, B.; PARKER, P.; SCOTT, R. A new strategy for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, v. 40, n. 1, p. 82–94, Jan 1993. ISSN 0018-9294. Citado na página 13.
- KATSIS, C. et al. A novel method for automated {EMG} decomposition and {MUAP} classification. *Artificial Intelligence in Medicine*, v. 37, n. 1, p. 55 – 64, 2006. ISSN 0933-3657. Intelligent Data Analysis in Medicine. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0933365705001065>>. Citado 2 vezes nas páginas 13 e 19.
- KLIN, J. C.; LUCA, C. J. D. Error reduction in emg signal decomposition. *Journal of Neurophysiology*, American Physiological Society, 2014. ISSN 0022-3077. Citado na página 16.
- LOPES, I. F. *Caracterização dos Sinais Mioelétricos dos Movimentos do Segmento Mão-Braço Através de Regressão Logística*. 2014. Trabalho de Diplomação, Curso de Engenharia de Computação, UFRGS. Citado na página 16.
- LUCA, C. J. D. et al. Decomposition of surface EMG signals. *J. Neurophysiol.*, v. 96, n. 3, p. 1646–1657, Sep 2006. Citado na página 15.
- PATTICHIS, C.; SCHIZAS, C.; MIDDLETON, L. Neural network models in emg diagnosis. *Biomedical Engineering, IEEE Transactions on*, v. 42, n. 5, p. 486–496, May 1995. ISSN 0018-9294. Citado 4 vezes nas páginas 13, 14, 17 e 21.
- SCHONS, L. *Caracterização dos Sinais Mioelétricos dos Movimentos do Segmento Mão-Braço Através de Redes Neurais Artificiais e Algoritmos Genéticos*. 2014. Trabalho de Diplomação, Curso de Engenharia de Computação, UFRGS. Citado 2 vezes nas páginas 16 e 31.

Apêndices

APÊNDICE A – Função em MATLAB para MTD1

```

function [x_seg, finalCenterLocs] = seg_mtd1(x, l, q, r_target, T_lim)
%   MTD1 - metodo iterativo utilizando thresholding para deteccao de
%   centros de segmentos de comprimento constante
%
% Argumentos: (para mais detalhes, refira a descricao do MTD1)
%   x - matriz cujas colunas sao canais do sinal a ser segmentado
%   l - comprimento desejado para os segmentos
%       (deve ser inteiro maior que zero)
%   q - razao de atualizacao entre iteracoes para valor de threshold
%       (deve ser entre 0 e 1)
%   r_target - razao minima esperada entre numero de segmentos e comprimento
%       total de sinal
%       (deve ser maior que zero)
%   T_lim - fracao do maximo do sinal para limite inferior de threshold
%       (deve ser entre 0 e 1)
%
% Retorno:
%   x_seg - cell array com canais segmentados
%   finalCenterLocs - posicoes centrais dos segmentos

%% Preprocessamento

[L, numberOfChannels] = size(x); % comprimento do sinal e numero de canais
x_ret = abs(x); % retificacao
x_norm = zeros(L, numberOfChannels); % normalizacao
for currentChannel = 1:numberOfChannels
    x_norm(:,currentChannel) = ...
        x_ret(:,currentChannel)./max(x_ret(:,currentChannel));
end

%% Metodo

centerLocsCell = cell(1,numberOfChannels);
for currentChannel = 1:numberOfChannels
    T_k = 1; % canais normalizados, seus valores maximos sempre sao 1
    targetReached = false;
    while ~targetReached % processo iterativo
        T_k = q*T_k; % calcula threshold desta iteracao
        if T_k < T_lim % verifica se o limite de valor de threshold foi atingido

```

```

        break
    end
    % Identifica candidatos a centros de segmentos
    [~, centerLocsCell{1,currentChannel}] = ...
        findpeaks(double(x_norm(:,currentChannel)), ...
        'MinPeakHeight', T_k, 'MinPeakDistance',1);
    % Determina o encerramento do processo iterativo
    targetReached = (length(centerLocsCell{1,currentChannel})/L >= r_target);
end
end

%% Clustering

centerLocsArray = sort(cell2mat(centerLocsCell'));
[~, labscore] = dbscan(centerLocsArray,2000,3);
numberOfSegments = max(labscore);
finalCenterLocs = zeros(numberOfSegments,1); % medias internas aos clusters
for currentCluster = 1:numberOfSegments
    finalCenterLocs(currentCluster) = ...
        round(mean(centerLocsArray(labscore == currentCluster)));
end

%% Segmentacao

x_seg = cell(numberOfSegments, numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        switch mod(1,2)
            case 0 % se l for par
                if(finalCenterLocs(currentSegment)-1/2)<1
                    % segmento muito a esquerda
                    x_seg{currentSegment,currentChannel} = ...
                        x(1:finalCenterLocs(currentSegment)+(1/2)-1, ...
                        currentChannel);
                else if(finalCenterLocs(currentSegment)+(1/2)-1)>L
                    % segmento muito a direita
                    x_seg{currentSegment,currentChannel} = ...
                        x(finalCenterLocs(currentSegment)-1/2:L, ...
                        currentChannel);
                else
                    x_seg{currentSegment,currentChannel} = ...
                        x(finalCenterLocs(currentSegment)-1/2: ...
                        finalCenterLocs(currentSegment)+(1/2)-1, ...
                        currentChannel);
                end
            end
            case 1 % se l for impar

```

```
if(finalCenterLocs(currentSegment) - (l-1)/2)<1
    % segmento muito a esquerda
    x_seg{currentSegment,currentChannel} = ...
        x(1:finalCenterLocs(currentSegment) + (l-1)/2, ...
            currentChannel);
else if(finalCenterLocs(currentSegment) + (l-1)/2)>L
    % segmento muito a direita
    x_seg{currentSegment,currentChannel} = ...
        x(finalCenterLocs(currentSegment) - (l-1)/2:L, ...
            currentChannel);
else
    x_seg{currentSegment,currentChannel} = ...
        x(finalCenterLocs(currentSegment) - (l-1)/2: ...
            finalCenterLocs(currentSegment) + (l-1)/2, ...
            currentChannel);
end
end
end
end
end
end
```

APÊNDICE B – Função em MATLAB para MTD2

```
function [x_seg, finalCenterLocs] = seg_mtd2(x, l, A, B, C)
% MTD2 - metodo nao iterativo utilizando thresholding para deteccao de
% centros de segmentos de comprimento constante
%
% Argumentos: (para mais detalhes, refira a descricao do MTD2)
% x - matriz cujas colunas sao canais do sinal a ser segmentado
% l - comprimento desejado para os segmentos
%      (deve ser inteiro maior que zero)
% A - coeficiente utilizado para decisao de metodo de calculo de threshold
%      (deve ser maior ou igual a 1)
% B - multiplo da media aritmetica do sinal x para obtencao de threshold
%      (deve ser maior ou igual a 1)
% C - fracao do valor maximo do sinal x para calculo de threshold
%      (deve ser maior ou igual a 1)
%
% Retorno:
% x_seg - cell array com os canais segmentados
% finalCenterLocs - posicoes centrais dos segmentos

%% Preprocessamento

[L, numberOfChannels] = size(x); % comprimento do sinal e numero de canais
x_ret = abs(x); % retificacao
x_norm = zeros(L, numberOfChannels); % normalizacao
for currentChannel = 1:numberOfChannels
    x_norm(:,currentChannel) = ...
        x_ret(:,currentChannel) ./ max(x_ret(:,currentChannel));
end

%% Metodo

centerLocsCell = cell(1,numberOfChannels);
for currentChannel = 1:numberOfChannels
    % Calculo do threshold
    maxValue = 1; % sinais normalizados
    meanValue = mean(x_norm(:,currentChannel));
    if maxValue > (A*meanValue)
        T = B*meanValue;
    else
```

```

        T = maxValue/C;
    end
    % Identifica centros de segmentos
    [~, centerLocsCell{1,currentChannel}] = ...
        findpeaks(double(x_norm(:,currentChannel)), ...
            'MinPeakHeight', T, 'MinPeakDistance',1);
end

%% Clustering

centerLocsArray = sort(cell2mat(centerLocsCell'));
[~, labscore] = dbscan(centerLocsArray,2000,3);
numberOfSegments = max(labscore);
finalCenterLocs = zeros(numberOfSegments,1); % medias internas aos clusters
for currentCluster = 1:numberOfSegments
    finalCenterLocs(currentCluster) = ...
        round(mean(centerLocsArray(labscore == currentCluster)));
end

%% Segmentacao

x_seg = cell(numberOfSegments, numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        switch mod(1,2)
            case 0 % se l for par
                if(finalCenterLocs(currentSegment)-1/2)<1
                    % segmento muito a esquerda
                    x_seg{currentSegment,currentChannel} = ...
                        x(1:finalCenterLocs(currentSegment)+(1/2)-1, ...
                            currentChannel);
                else if(finalCenterLocs(currentSegment)+(1/2)-1)>L
                    % segmento muito a direita
                    x_seg{currentSegment,currentChannel} = ...
                        x(finalCenterLocs(currentSegment)-1/2:L, ...
                            currentChannel);
                else
                    x_seg{currentSegment,currentChannel} = ...
                        x(finalCenterLocs(currentSegment)-1/2: ...
                            finalCenterLocs(currentSegment)+(1/2)-1, ...
                            currentChannel);
                end
            end
            case 1 % se l for impar
                if(finalCenterLocs(currentSegment) - (1-1)/2)<1
                    % segmento muito a esquerda
                    x_seg{currentSegment,currentChannel} = ...

```

```
        x(1:finalCenterLocs(currentSegment) + (l-1)/2, ...  
          currentChannel);  
    else if(finalCenterLocs(currentSegment) + (l-1)/2)>L  
        % segmento muito a direita  
        x_seg{currentSegment,currentChannel} = ...  
            x(finalCenterLocs(currentSegment) - (l-1)/2:L, ...  
              currentChannel);  
    else  
        x_seg{currentSegment,currentChannel} = ...  
            x(finalCenterLocs(currentSegment) - (l-1)/2: ...  
              finalCenterLocs(currentSegment) + (l-1)/2, ...  
              currentChannel);  
    end  
end  
end  
end  
end  
end  
end
```

APÊNDICE C – Função em MATLAB para grupamento com DBSCAN

```

function [labs labscore] = dbscan(a,Eps,MinPts)
% DBSCAN clustering
% [labs labscore] = dbscan(A,Eps,MinPts)
%
% DBSCAN clustering of data matrix in A. labels is a vector with
% cluster labels for each vector.
%
% In case of publication of any application of this method,
% please, cite the original work:
% Thanh N. Tran*, Klaudia Drab, Michal Daszykowski, "Revised DBSCAN algorithm
% to cluster data with dense adjacent clusters", Chemometrics and Intelligent
% Laboratory Systems, 120:92-96.
% DOI: 10.1016/j.chemolab.2012.11.006

UNCLASSIFIED = 0;
BORDER = -2;

% Square Eps in order not to square all distances of points
%eps = eps^2;

m = size(a,1);
labs = zeros(m,1);
ClusterId = 1;
for i=1:m
    if labs(i) == UNCLASSIFIED

        % Expand cluster ClusterId
        % Get a set of points of distance < eps
        [ExpandClusterReturn labs]= ExpandCluster(a,labs,i,ClusterId,Eps, MinPts);
        if ExpandClusterReturn
            ClusterId = ClusterId +1;
        end
    end
end

% Step 3:
labscore = labs; core_index = find(labscore > 0);
border_points = find(labs==BORDER);
% For xborder in border_list but has no ClusterId

```

```

    for i=1:length(border_points)
        % xborder in border_list
        currentB = border_points(i);
        d=distance(+a(currentB,:),+a(core_index,:),1);
        % the closest core-points
        [tmp nearest_core]=min(d);
        nearest_core_index=core_index(nearest_core);
        %Assign xborder to ClusterId of the closest core-points
        labs(currentB)=labs(nearest_core_index);
    end
end

function [ExpandClusterReturn labs]= ExpandCluster(a,labs,i,ClusterId, Eps, MinPts)
UNCLASSIFIED = 0;
NOISE = -1;
BORDER = -2;
    % calculate distances
    d=distance(+a(i,:),+a(:,,:),1);

    % seeds = Retrieve_Neighbors(xi, Eps)
    seeds = find(d < Eps);

    % If |seeds| < MinPts
    if size(seeds,2) < MinPts,
        labs(i) = NOISE; % Assign xi as noise
        ExpandClusterReturn = 0; % Return without expansion success
    else
        % STEP 1: xi is identified as a starting core-point for ClusterId
        %labs(i) = ClusterId; % Thanh changed in May 2012
        % Exclude xi from the Seeds
        %seeds = setdiff(seeds, i); % Thanh changed in May 2012
        % STEP 2: Identify chains
        % For all xj in seeds
        while ~isempty(seeds) % Not an empty seeds
            % current point is the first point in seeds

            currentP = seeds(1);
            d=distance(+a(currentP,:),+a(:,,:),1);
            % NEps(xj)= Retrieve_Neighbors(xj,Eps)
            result = find(d <= Eps);
            %If | NEps(xj) | >= MinPts

            // xj is a core point
            if length(result) >= MinPts
                % Assign xj to ClusterId
                labs(currentP) = ClusterId;
                % Add all UNCLASSIFIED in NEps(xj) to seeds
                result_unclassified = result(find(labs(result)==UNCLASSIFIED));
            end
            % Remove currentP from seeds
            seeds = setdiff(seeds, currentP);
        end
    end
    ExpandClusterReturn = labs(i);
end

```



```
        result_noise = result(find(labs(result)==NOISE));
        % Temporary complete the intermediate chain ... the chain can
        % The border points have not yet been assigned to any cluster
        labs([result_unclassified result_noise]) = BORDER;
    % first assign to a border-point... later will be reassigned to e.g. core-point
        seeds = union(seeds,result_unclassified);
    end
    % Exclude the current point in seeds and go back to the loop
    seeds = seeds(2:size(seeds,2));
end % end while
% Return with expansion success
ExpandClusterReturn = 1;    % return true
end
end
```

APÊNDICE D – Código utilizado para determinação de r_{target} do MTD1

```
%%
%   Determinacao de r_target para MTD1
%%

%% Base de dados Ninapro

ninaproList = ls('database/ninapro2/S*_E1*');
numberOfSubjects = length(ninaproList);
numberOfSegments = 102; % 17 movimentos * 6 repeticoes

%% Obtencao de r_target da database

r_target = zeros(numberOfSubjects, 1);

for currentSubject = 1:numberOfSubjects

    fprintf('currentSubject = %i\n',currentSubject)

    % Carrega o voluntario atual
    load(['database/ninapro2/' ninaproList(currentSubject,:)])

    % r_target para este voluntario
    r_target(currentSubject) = numberOfSegments/length(emg);

end

min_r_target = min(r_target)
```

APÊNDICE E – Função em MATLAB para obtenção dos movimentos relacionados aos segmentos

```
function targetDataArray = identifyClasses(centerLocs, stimulus)

%% Remodela o vetor de estímulo, removendo trechos de 0

reshapedStimulus = stimulus;
currentMovement = 1;
for index = 1:length(stimulus)
    if reshapedStimulus(index) ~= 0
        currentMovement = reshapedStimulus(index);
    else
        reshapedStimulus(index) = currentMovement;
    end
end

%% Gera a matriz de identificacao das classes

numberOfSegments = length(centerLocs);
numberOfClasses = 17;
targetDataArray = false(numberOfSegments, numberOfClasses);
for currentSegment = 1:numberOfSegments
    targetDataArray(currentSegment,...
        reshapedStimulus(centerLocs(currentSegment))) = true;
end

end
```